

基于 t-SNE 算法的 ICN 缓存容量分配机制

陈香伊¹, 王兴伟¹, 李 婕¹, 易 波¹, 黄 敏²

(1. 东北大学 计算机科学与工程学院 辽宁 沈阳 110169; 2. 东北大学 信息科学与工程学院 辽宁 沈阳 110819)

摘要: 针对现有 ICN 中节点负载不均衡、重要节点缓存利用率低、缓存开销大等问题, 面向互联网主干网节点, 提出了基于 t-SNE 算法的 ICN 缓存容量分配机制。首先, 收集网络拓扑信息和流量特征信息并建立高维数据集; 然后, 通过构造 K-近邻表征相似性的方式改进 t-SNE 算法, 对数据集进行降维并对网络节点进行聚类划分; 最后, 基于聚类结果, 将有限的缓存容量合理地分配给不同节点以平衡节点负载。仿真结果表明, 本设计的缓存容量分配机制和基准机制相比, 在保证路由成功率维持在约 95% 的前提下, 缓存命中率提升了 3%~4%, 平均缓存开销减少了 13.5%~23.4%。

关键词: 信息中心网络; 容量分配; t-SNE 算法; 负载均衡

中图分类号: TP393

文献标志码: A

文章编号: 1671-6841(2019)03-0022-06

DOI: 10.13705/j.issn.1671-6841.2018331

0 引言

随着互联网数据的快速增长和网络应用的日趋丰富, 用户需求逐渐从主机之间的通信演进为主机对网络信息的重复访问^[1-2]。与传统网络架构中以 IP 地址进行路由的方式不同, 信息中心网络(information-centric networking, ICN)^[3-4]通过唯一的内容名称对用户请求进行路由, 且每个节点除了具有处理、转发的功能之外, 还具有存储的功能^[5]。网内缓存作为 ICN 最大特点之一^[6-7], 在提高用户服务质量、减少用户访问时延、减轻服务器负载上功不可没^[8]。ICN 缓存领域中很多关键技术已有了阶段性的创新, 但仍值得深入分析和研究^[3]。

在 ICN 众多研究项目中, 最具代表性且最有发展前景的范例当属命名数据网络(named data networking, NDN)项目^[9-10]。文献[11]为了充分利用 ICN 的内置缓存, 提出了一种基于内容空间分区和哈希路由的缓存机制, 将内容缓存在指定的划分区域中, 能够解决哈希路由引起的路径拉伸问题。文献[12]提出了一种新型智能资源管理系统, 旨在分析请求模式, 充分利用通用缓存内容。该系统能够根据用户需求变化实时高效地进行缓存资源分配。文献[13]通过在转发信息库中添加路由缓存, 包括原子缓存和即时缓存, 来缓解转发信息库的爆炸问题。目前国内外学者在 ICN 体系结构、路由算法、缓存决策等方面已经取得了一定的成果, 但是却鲜有针对缓存容量分配机制的研究。

ICN 缓存容量分配面临着巨大的缓存对象与有限的缓存空间之间的矛盾^[14]。绝大多数的 ICN 中都默认均匀部署路由器(即缓存容量相同)。考虑到节点位置、网内流量分布以及用户请求特征不同, 从根本上导致了节点负载不均衡。因此, 如何将有限的总缓存容量适当地部署到更关键的位置, 以平衡节点负载并提高缓存利用率尤为关键。

针对上述问题, 本文在互联网主干网络的 ICN 网络架构下, 分别从用户角度和节点角度对全局流量分布进行分析。首先分析网络拓扑属性以及流量特征信息, 然后基于分析结果对网络节点进行聚类划分, 并重新分配节点容量, 旨在将有限的总缓存容量以最优的方式合理地分配给各节点, 从而实现网络的负载均衡。

收稿日期: 2018-12-20

基金项目: 国家自然科学基金项目(61572123, 61872073); 辽宁省高校创新团队支持计划项目(LT2016007); 教育部-中国移动科研基金项目(MCM20160201)。

作者简介: 陈香伊(1994—), 女, 四川内江人, 硕士研究生, 主要从事未来互联网、云计算研究, E-mail: chenxiangyi8@163.com; 通信作者: 王兴伟(1968—), 男, 辽宁盖州人, 教授, 主要从事未来互联网、云计算、网络空间安全研究, E-mail: wangxw@mail.neu.edu.cn。

1 网络数据模型

1.1 拓扑信息

网络拓扑中的节点位置从根本上影响着该节点处理兴趣请求的概率,即位于网络中相对“枢纽”位置的节点可能需要处理更多的请求.因此,基于图论基础,选取了节点度数中心性 C_d 、节点介数中心性 C_b 、节点紧密中心性 C_c 3 个中心性指标作为网络拓扑属性来区分节点的重要程度.

节点度数中心性 C_d 的计算为 $C_{d_i} = \sum_{k=1}^n a(v_i, v_k)$, 其中: n 是节点总数; $a(v_i, v_k)$ 是连接函数.如果节点 v_i, v_k 直接相连,那么 $a(v_i, v_k)$ 值为 1, 否则为 0; C_d 表示与该节点直接相连的边数,用来分析节点的直接

影响力.节点介数中心性 C_b 的计算为 $C_{b_i} = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(i)}{\sigma_{st}}$, 其中: σ_{st} 是点 v_s 到点 v_t 的所有最短路径数; $\sigma_{st}(i)$ 是其中经过节点 i 的路径数,节点介数中心性表示网络中所有最短路径中经过该节点的路径的数目占最短路径总数的比例,反映了该节点是否为“交通枢纽”,即在整个网络中的作用和影响力.节点紧密中心性

C_c 的计算为 $C_{c_i} = \frac{1}{\sum_{k=1}^n d(v_i, v_k)}$, 其中: $d(v_i, v_k)$ 是节点 v_i 到节点 v_k 的最短路径长度.一个节点如果到其他

所有节点的最短路径和越小,则该点就越靠近网络的中心位置,用来分析节点的相对位置.

1.2 流量特征信息

流量特征信息能够很好地反映网络流量的分布情况,为区分节点的重要程度.本文从节点负载和用户偏好两方面共选取 5 个指标作为流量特征信息属性.在节点负载方面,选取接收兴趣包个数 Rec_I 、响应请求次数 Res_I 以及内容替换次数 Rep_C 3 个指标.在用户偏好方面,选取兴趣聚合率 $Aggr$ 和兴趣影响度 $Impact$ 两个指标.其中,节点负载方面的 3 个指标可以通过采样单位时间段内的测量值累加统计得出.下面给出本文对兴趣聚合率和兴趣影响度的定义.

兴趣聚合率 $Aggr$ 为节点 v_i 添加的接口数 $Nface_i$ 与节点接收到的兴趣包数 Rec_I_i 的比值,如 $Aggr_i = \frac{Nface_i}{Rec_I_i}$. 兴趣影响度 $Impact_i = \sum_{j=0}^{TTL} \left(Ihop_j \cdot \frac{1}{j+1} \right)$. TTL 为用户所能容忍的最大跳数.本文定义节点 v_i 响应的兴趣请求中,兴趣包经过跳数 j 的倒数和对应的兴趣包个数 $Ihop_j$ 之积的总和为兴趣影响度.

1.3 信息整合模型

本文将提出的 3 个拓扑参数和 5 个流量信息参数作为评价节点重要程度的指标.使用 $x_i(t)$ 表示,在单位取样时间段 T 内,节点 v_i 收集到的流量特征信息的数据,如公式(1),

$$x_i(T) = \{ Rec_I_i(T), Res_I_i(T), Rep_C_i(T), Aggr_i(T), Impact_i(T) \}, \quad (1)$$

其中: $x_i(T)$ 是五维数据,简称为 x_i .单独收集一个时间段内的流量信息不能全面反映网内流量的变化情况,因此,总共需要选取 T' 个采样时间段进行流量信息的收集.用 X_i 表示在 T' 个采样时间段内节点 v_i 收集到的数据集为

$$X_i = \{ x_i(T_1), x_i(T_2), \dots, x_i(T_{T'}) \}, C_{d_i}, C_{b_i}, C_{c_i} \quad i \in [0, n). \quad (2)$$

结合公式(1),可将公式(2)中的 X_i 展开

$$X_i = \{ Rec_I_i(T_1), Res_I_i(T_1), Rep_C_i(T_1), Aggr_i(T_1), Impact_i(T_1), \dots, Rec_I_i(T_{T'}), Res_I_i(T_{T'}), Rep_C_i(T_{T'}), Aggr_i(T_{T'}), Impact_i(T_{T'}), C_{d_i}, C_{b_i}, C_{c_i} \},$$

其中: X_i 是 $(5T' + 3)$ 维的数据.考虑到拓扑中共有 n 个节点,因此,本文收集的数据集为 n 个 $(5T' + 3)$ 维的数据.

2 缓存容量分配机制

2.1 基于改进 t-SNE 算法的数据集处理

本文所收集的数据呈现高维特征,且对于每个节点而言高维数据之间存在数据冗余、信息重叠的问题.

本文将使用改进的 t-SNE 算法^[15]对原数据进行降维,并通过 VP 树(vantage point tree)^[16]构建 K -近邻.在高维空间中,已知 x_i 和 x_j 为任意的两个数据点, x_k 为非 x_i 的数据点, Nei_i 为 x_i 邻居集合,可通过对称化两个条件概率分布得到点对相似性的联合概率分布 p_{ij} .而在本文收集的高维数据点中,两个相距较远的数据点的相似性概率 p_{ij} 非常小,因此取条件概率作为 p_{ij} 的近似.基于此,高维空间中数据点的点对相似性可定义为

$$p_{ji} = \begin{cases} \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \in Nei_i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}, & \text{如果 } j \neq i \\ 0, & \text{其他} \end{cases} \quad (3)$$

高维数据是基于流形的^[17],在高维空间中如果直接使用欧式距离会存在误差,因此本文将以测地线距离代替欧式距离.但是,高维数据中的真实测地线距离难以获得.考虑到每个输入对象 x_i 已有最近邻居集合 Nei_i ,本文将邻域内两个数据点间的最短路径计算值作为真实测地线距离的近似. x_i 邻域 Nei_i 内任意两个数据点间的欧式距离记为 $d_x(x_i, x_j)$,测地线距离记为 $d_c(x_i, x_j)$,两者关系如公式(4),然后计算 x_i 邻域 Nei_i 内任意两个数据点间的最短路径,如公式(5),

$$d_c(x_i, x_j) = \begin{cases} d_x(x_i, x_j), & \text{如果 } i, j \text{ 相连} \\ \infty, & \text{其他} \end{cases} \quad (4)$$

$$d_c(x_i, x_j) = \min\{d_c(x_i, x_j), d_c(x_i, x_k) + d_c(x_k, x_j)\}, k \in [0, n]. \quad (5)$$

基于此,公式(3)可改进为

$$p_{ji} = \begin{cases} \frac{\exp(-(x_i, x_j)^2/2\sigma_i^2)}{\sum_{k \in Nei_i} \exp(-d_c(x_i, x_k)^2/2\sigma_i^2)}, & \text{如果 } j \neq i \\ 0, & \text{其他} \end{cases} \quad (6)$$

然后使用 KL 散度(Kullback-Leibler divergence)来衡量联合概率分布 p_{ij} 和 q_{ij} 之间的相似性.t-SNE 最终

目标就是对高维空间中所有数据点最小化代价函数 $C, C = KL(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$.在初始化的过程中,方差 σ 可以选择较小的值,使用随机梯度下降算法进行训练.在训练过程中,为了保证优化速度,同时防止陷入局部最优,需要引入 3 个参数,分别为迭代次数 I 、学习速率 η 和第 i 次迭代的动量参数 $\alpha(i)$.在更新过程中,不仅需要考虑当前的梯度,还需要考虑之前迭代过程中梯度的指数衰减.根据文献[16]推导,迭代 i 次后的解 $Y^{(i)} = Y^{(i-1)} + \eta \frac{\delta C}{\delta Y} + \alpha(i)(Y^{(i-1)} - Y^{(i-2)})$.经过 I 次迭代后,得到低维嵌入的最终解 $Y^{(I)}$.

2.2 缓存容量分配

在通过构造 K -近邻表征相似性的方式改进了 t-SNE 算法后,对收集到的高维数据进行处理,得到低维空间的嵌入结果 $Y^{(I)}$.本文以该降维结果作为节点相似性的划分标准,根据节点重要程度不同,设置不同的权重,进而按权重为节点分配不同大小的容量空间.

假设低维嵌入结果 $Y^{(I)}$ 将节点划分为 k_class 类, num_j 表示类别 j 中包含的节点个数.令 W 表示类别对应的权重值, $W = \{w_1, w_2, \dots, w_k\}$, W 中的变量按降序排列且满足公式(7)的约束,

$$\sum_{j=1}^k w_j = 1, 0 < w_1 < \dots < w_k < 1. \quad (7)$$

假设整个网络拓扑的总缓存容量用 C_{total} 表示,在类别 j 中,节点的分配缓存大小用 C_{ji} 表示,

$$C_{ji} = C_{total} \cdot w_j / num_j, j \in [1, k], i \in [1, num_j]. \quad (8)$$

缓存容量分配机制中,首先初始化参数并构建 VP 树;然后根据 VP 树构建 K -近邻并计算点对之间的测地线距离;再计算高维空间联合概率分布;最后使用梯度下降法进行训练,得到低维嵌入的结果.通过构造 K -近邻表征相似性的方式改进 t-SNE 算法,既考虑了高维数据的流形特征,也大大降低了高维空间点对相似性的计算量.

3 性能评价

3.1 拓扑用例

在进行仿真实验的过程中,本文使用了中国的 cernet 和美国的 deltagcom 两种拓扑结构. cernet 网络拓扑结构含有 36 个节点、49 条链路,平均节点度数为 2.72. deltagcom 网络拓扑结构含有 113 个节点、161 条链路,平均节点度数为 2.85.

3.2 评价指标

本文在 Ubuntu 下搭建基于 NS-3 的仿真模块 NDN-SIM 并进行仿真实验,同时将搜集的历史流量数据导入到 matlab 软件中进行处理.

本文基于两种拓扑结构,采用多个性能指标对基于 t-SNE 算法的缓存容量分配机制进行性能评价,评价指标如下.

1) 缓存命中率. 缓存命中率的计算公式 $CacheHitRatio = Num_{cacheHit} / Num_{success}$, 其中: $Num_{cacheHit}$ 为从路由器节点的内容存储库 (content store, CS) 获得内容的兴趣请求数; $Num_{success}$ 为路由成功的兴趣请求数. 缓存命中率能够在很大程度上表征缓存内容的利用率.

2) 路由成功率. 路由成功率的计算公式 $SuccessRatio = Num_{success} / Num_{total}$, 其中: $Num_{success}$ 为路由成功的兴趣请求数; Num_{total} 为用户向网络中发起的兴趣请求总数. 路由成功率能够表征路由机制的效率.

3) 缓存开销. 节点缓存开销的计算公式 $Over_i = Rep_{C_i} \cdot Vol_{aver} / c_i$, 其中: Rep_{C_i} 为节点内容替换次数; Vol_{aver} 为内容平均大小; c_i 为节点缓存容量. 缓存开销能够表征采样时间段内节点的负载情况.

3.3 性能评估

为验证本文设计的基于 t-SNE 算法的缓存容量分配机制,在全网发起了 5 000 次请求进行节点聚类,所有节点都能正常执行缓存替换机制 (缓存空间已满) 并收集 50 个采样时间段的流量特征信息.

在 cernet 拓扑下执行本文的缓存容量分配机制得到的二维嵌入结果如图 1 所示,其中每个数据点代表一个路由器节点. 根据节点在网络中的相对位置可以发现, X 值越大, Y 值越小,节点在网络中越重要. 由该图可见 cernet 的 36 个节点被聚簇为 2 类,含有重要节点 8 个,含有普通节点 28 个. 在 deltagcom 拓扑下执行本文的缓存容量分配机制得到的二维嵌入结果如图 2 所示. deltagcom 中的 113 个节点被聚簇为 4 类,其中节点个数分别为 6、40、30、37.

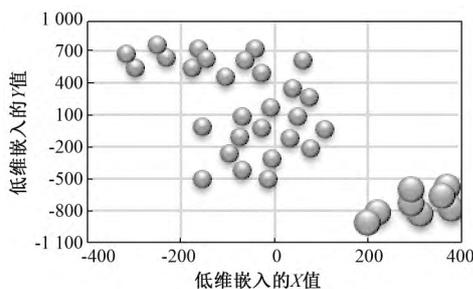


图1 cernet 拓扑下的低维嵌入结果

Fig.1 The low dimension result in cernet

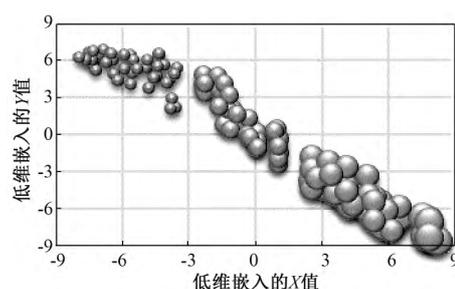


图2 deltagcom 拓扑下的低维嵌入结果

Fig.2 The low dimension result in deltagcom

为了验证本文提出的缓存容量分配机制的性能,选取 LCE (leave copy everywhere) 以及 Betw^[18] 作为基准缓存决策机制. 其中, LCE 是 ICN 的默认缓存决策机制,它在数据包传输路径上的每个路由器节点都进行缓存;而 Betw 机制对 LCE 进行了改进,它将内容缓存在兴趣包转发路径上介数中心性最大的节点中. 本文在 cernet 和 deltagcom 两种网络拓扑下,将两种基准机制分别在默认等量分配和基于 t-SNE 算法的缓存容量分配 (t-LCE、t-Betw) 两种情况下进行仿真实验. 在网内发起 3 000 个兴趣请求,并统计对应缓存命中率、路由成功率、缓存开销 3 个指标. 实验结果如表 1 所示.

从表 1 中我们可以看出,在引入缓存容量分配机制后, t-LCE 与 t-Betw 均在不同程度上提升了基准缓存机制的缓存命中率和路由成功率,且降低了网络平均缓存开销. 本文提出的缓存容量分配机制基于低维嵌入

表 1 Cernet 与 delatcom 两种拓扑下对比实验结果

Tab.1 Comparative experimental results over cernet and delatcom

%

评价指标	cermet 拓扑			delatcom 拓扑		
	缓存命中率	路由成功率	平均缓存开销	缓存命中率	路由成功率	平均缓存开销
LCE	75.27	86.60	44.50	80.20	94.10	35.29
t-LCE	79.13	88.24	38.20	84.23	95.53	21.79
Betw	76.83	87.53	48.11	78.83	93.47	32.88
t-Betw	77.47	90.69	24.71	81.76	97.39	15.48

的结果,为节点分配不同大小的缓存容量.cernet 拓扑下,均匀分布时每个节点缓存容量为 1 G,重要节点权重 $w_1 = 0.6$,普通节点权重 $w_2 = 0.4$,基于低维嵌入结果,每个重要节点需分配 2.7 G 容量,每个普通节点需分配 0.5 G 容量($2.7 \text{ G} \times 8 + 0.5 \text{ G} \times 28 \approx 36 \text{ G}$).delatcom 拓扑下,同样均匀分布时节点缓存容量为 1 G, A 类节点权重分别设为 $w_1 = 0.4, w_2 = 0.3, w_3 = 0.2, w_4 = 0.1$.根据公式(8)可知 A 类节点分别需要分配 7.5 G、0.85 G、0.75 G、0.3 G 容量($7.5 \text{ G} \times 6 + 0.85 \text{ G} \times 40 + 0.75 \text{ G} \times 30 + 0.3 \text{ G} \times 37 \approx 113 \text{ G}$).

在两种拓扑下,本文提出的缓存容量分配机制与默认机制进行对比分析,缓存命中率提高了 3%~4%,路由成功率基本维持在 95%,其性能的提升与缓存决策机制本身有关.t-LCE 机制对缓存命中率的提升更明显.因为从全网角度来看,尽管普通节点容量减少,但重要节点分配更多的容量后,不仅特别流行的内容副本数变化不大,还能够在重要节点缓存更多的流行内容,因此在缓存命中率指标上提升很多.t-Betw 机制则在路由成功率提升方面更加明显,这是由于该机制在当前路径中的重要节点缓存内容,缓存容量分配机制为网络中相对重要的节点分配更多的缓存容量,这些节点能够响应更多的兴趣请求,因此极大提升了路由成功率.

从全网节点的平均缓存开销来看,cernet 拓扑下 t-LCE 的平均缓存开销减少了 6.3%,t-Betw 的平均缓存开销减少了 23.4%.delatcom 拓扑下 t-LCE 的节点平均开销减少了 13.5%,t-Betw 的节点平均开销减少了 17.4%.由此可见,本文提出的缓存容量分配机制在 Betw 上提升效果更明显,这是因为 Betw 对节点按照节点介数中心性进行划分,而不是像 LCE 机制那样对所有节点一视同仁.此外,本文提出的缓存容量分配机制,权衡了拓扑信息和流量分布,将节点按照重要程度进行划分,进而能够保证网络流量按照节点重要程度均匀分布在网内,不仅解决了部分节点负载过重的问题,还解决了少数节点缓存利用率不高的问题,从而降低了网络的整体缓存开销.

4 结束语

本文通过对现有缓存技术进行分析,提出了基于 t-SNE 算法的缓存容量分配机制.通过构造 K-近邻表征相似性的方式改进 t-SNE 算法,对网络拓扑属性以及流量特征信息进行分析,基于聚类结果将有限的缓存空间合理分配给不同节点以达到平衡节点负载的目的.为验证本文方法的可行性和有效性,对算法进行了仿真实现,并与基准机制进行对比分析.从实验结果可以看出,本文设计的机制在缓存命中率、路由成功率以及缓存开销等方面具有一定的优势.下一步工作将对缓存分配机制的稳定性进行验证,进一步完善本文提出的机制.

参考文献:

- [1] ZHANG Z, LUNG C H, LAMBADARIS I, et al. When 5G meets ICN: an ICN-based caching approach for mobile video in 5G networks[J]. Computer communications, 2018, 118: 81-92.
- [2] ABDULLAHI I, ARIF S, HASSAN S. Survey on caching approaches in information centric networking[J]. Journal of network and computer applications, 2015, 56: 48-59.
- [3] XYLOMENOS G, VERVERIDIS C N, SIRIS V A, et al. A survey of information-centric networking research[J]. IEEE communications surveys & tutorials, 2014, 16(2): 1024-1049.
- [4] VASILAKOS A V, ZHE L, SIMON G, et al. Information centric network: research challenges and opportunities[J]. Journal of network and computer applications, 2015, 52: 1-10.

- [5] ZHANG G Q , LI Y , LIN T. Caching in information centric networking: a survey [J]. *Computer networks* , 2013 , 57(16) : 3128–3141.
- [6] ZHANG M , LUO H , ZHANG H. A survey of caching mechanisms in information–centric networking [J]. *IEEE communications surveys & tutorials* , 2015 , 17(3) : 1473–1499.
- [7] KIM D , KIM Y. Enhancing NDN feasibility via dedicated routing and caching [J]. *Computer networks* , 2017 , 126: 218–228.
- [8] PSARAS I , CHAI W K , PAVLOU G. In–network cache management and resource allocation for information–centric networks [J]. *IEEE transactions on parallel and distributed systems* , 2014 , 25(11) : 2920–2931.
- [9] MAURI G , GERLA M , BRUNO F , et al. Optimal content prefetching in NDN vehicle–to–infrastructure scenario [J]. *IEEE transactions on vehicular technology* , 2017 , 66(3) : 2513–2525.
- [10] KARAMI A , GUERRERO–ZAPATA M. An ANFIS–based cache replacement method for mitigating cache pollution attacks in named data networking [J]. *Computer networks* , 2015 , 80: 51–65.
- [11] WANG S , BI J , WU J P , et al. CPHR: in–network caching for information–centric networking with partitioning and hash–routing [J]. *IEEE/ACM transactions on networking* , 2016 , 24(5) : 2742–2755.
- [12] ZHANG H , ZHU S , XIE R , et al. Intelligent resources management system design in information centric networking [J]. *China communications* , 2017 , 14(8) : 105–123.
- [13] CHEN X , ZHANG G Q , CUI H J. Investigating route cache in named data networking [J]. *IEEE communications letters* , 2018 , 22(2) : 296–299.
- [14] KUROSE J. Information–centric networking: the evolution from circuits to packets to content [J]. *Computer networks* , 2014 , 66: 112–120.
- [15] MAATEN L V D , HINTON G. Visualizing data using t–SNE [J]. *Journal of machine learning research* , 2008 , 9: 2579–2605.
- [16] KRYSZKIEWICZ M , JANČZAK B. Basic triangle inequality approach versus metric VP–tree and projection in determining euclidean and cosine neighbors [M]. Switzerland: Springer , 2014: 27–49.
- [17] BALASUBRAMANIAN M , SCHWARTZ E L. The isomap algorithm and topological stability [J]. *Science* , 2002 , 295(5552) : 7–11.
- [18] CHAI W , HE D , PSARAS I , et al. Cache “less for more” in information–centric networks (extended version) [J]. *Computer communications* , 2013 , 36(7) : 758–770.

ICN Cache Capacity Allocation Mechanism Based on t-SNE Algorithm

CHEN Xiangyi¹ , WANG Xingwei¹ , LI Jie¹ , YI Bo¹ , HUANG Min²

(1. *School of Computer Science and Engineering , Northeastern University , Shenyang 110169 , China;*

2. *School of Information Science and Engineering , Northeastern University , Shenyang 110819 , China)*

Abstract: Aiming at the status quo of unbalanced node load , low cache utilization of critical nodes as well as high cache overhead , an ICN cache capacity allocation mechanism based on t-SNE algorithm for internet backbone nodes was proposed. Firstly , high–dimensional data sets were established by collecting the network topology attributes and the traffic feature information. Secondly , dimension reduction was executed , and network nodes were clustered by constructing the similarity of K -nearest neighbors to improve the t-SNE algorithm. Finally , based on the clustering results , the limited cache capacity was allocated to different nodes reasonably to balance the node load. The simulation results showed that the cache capacity allocation mechanism designed increased the cache hit ratio by nearly 3%~5% , and reduced the average cache overhead by 13.5%~23.4% , while ensuring the routing success rate at 95% compared with the benchmark mechanisms.

Key words: information–centric networking; capacity allocation; t-SNE algorithm; load balancing

(责任编辑: 王浩毅)